A Multi-Stage Agent Framework with Knowledge Graph Alignment for Natural Product Literature Extraction

Shixiong Zhao[2,1,*], Yanming He[2,1], Lakshan Karunathilake[2,1] and Hideaki Takeda[1,2,*]

[1]National Institute of Informatics, Tokyo, Japan
[2]The Graduate University for Advanced Studies, SOKENDAI, Hayama, Japan

Abstract

Natural product literature contains extensive biochemical knowledge distributed across full-length scientific articles. Extracting this knowledge into structured representations requires robust handling of OCR text, long-document context, multilingual terminology, and entity normalization against controlled knowledge resources. We present A Multi-Stage Agent Framework with Knowledge Graph Alignment for Natural Product Literature Extraction, a document-level framework for extracting five core attributes from natural product articles: compound name, bioactivity, source species, collection site, and type of isolation. The framework begins with OCR over raw PDF articles and converts the documents into machine-readable full text. FullTextExtractor then performs article-level extraction over the full document and produces candidate values together with compound-bioactivity correspondences. A property-isolated knowledge graph retrieval module searches candidate entities from the corresponding knowledge graph for each extracted value. KGEntityNormalizer subsequently conducts value-level alignment and generates normalized outputs through preservation, completion, or canonical entity substitution according to the evidence in the article and the retrieved graph entities. This multi-stage design integrates document-level reading, structured candidate retrieval, and knowledge graph alignment into a unified extraction workflow. Experimental results on the BiKE test setting show that the proposed framework achieves strong performance across the five target attributes, with particularly reliable normalization for species, collection sites, and isolation types, demonstrating the effectiveness of multi-stage agent coordination with property-specific knowledge graph alignment for natural product literature mining.

Keywords

Biochemical knowledge extraction, Natural products, Large language models, Knowledge graph alignment, Entity normalization

1. Introduction

Scientific literature on natural products contains detailed information about compounds, source organisms, collection contexts, and biological activities. When this information is structured as a biochemical knowledge graph, it can support search, curation, biodiversity analysis, and downstream discovery. The relevant facts are often distributed across article titles, abstracts, experimental sections, tables, figure captions, and supplementary descriptions. Manual curation is accurate and slow, and the continued growth of the literature makes fully manual maintenance difficult.

The Biochemical Knowledge Extraction (BiKE) challenge addresses this setting by asking participants to extract biochemical properties from research articles and reconstruct hidden links in the NatUKE benchmark knowledge graph. The target properties are compound name, bioactivity, source species, collection site, and type of isolation. BiKE 2025 demonstrated that large language models are strong baselines for this task. In particular, the winning system used proprietary LLMs and an ensemble strategy to reach state-of-the-art performance across the benchmark properties [1]. The BiKE 2025 preface also emphasized the challenge's broader aim: turning biochemical article evidence into structured knowledge graph content [2].

That success creates a practical design question for future systems. If an LLM can extract the five properties when given a full paper and a rich prompt, the knowledge graph information still needs an effective organization strategy. Candidate-rich prompts constrain output values, improve matching,

BiKE'26: Third International Biochemical Knowledge Extraction Challenge, May 10–May 14, 2026, co-located with the 23rd European Semantic Web Conference (ESWC), Dubrovnik, Croatia
*Corresponding authors.

but they also introduce token cost, context pressure, and implementation complexity. These pressures become stronger when candidate sets are large, properties have different normalization rules, and paper evidence must be interpreted alongside graph-side hints.

This paper presents a submitted BiKE'26 system built around a modular pipeline. The pipeline separates article reading, knowledge graph retrieval, and entity normalization. The first LLM reads the OCR text of a full article and proposes values for all five properties. A local verifier then retrieves candidates from one graph per property. These graphs are built from the graph resources allowed before test evaluation and do not include hidden test labels. This design keeps compound names compared with compound names, species with species, and so on. A second LLM receives the extraction output together with the retrieved same-property candidates and emits one decision for each extracted value.

The contributions of this work are:

- a multi-stage LLM framework for extracting biochemical attributes from full-text natural product articles;
- a task decomposition strategy that separates article reading, KG candidate retrieval, and entity normalization, allowing the system to use KG evidence without placing large entity lists directly in the extraction prompt;
- an empirical analysis showing strong performance on the BiKE test setting, including near-SOTA compound name extraction and SOTA-level isolation type extraction.

2. Background and Motivation

NatUKE was introduced as a benchmark for extracting natural product knowledge from academic literature [3]. The benchmark combines scientific articles, manually curated biochemical properties, and graph structures that support link-prediction-style evaluation. Earlier systems relied heavily on graph representation learning and related embedding methods. The recent BiKE winner shifted the emphasis toward LLM-based direct extraction: full article text and candidate information were placed into prompts for models such as GPT-4.1 and Gemini 2.5, and the final ensemble improved substantially over graph embedding baselines [1]. The same paper also noted limitations of single-prompt approaches, including cost, output formatting, and the burden of including possible targets in prompts.

The submitted system is motivated by the information load observed in the challenge setting. A full-paper prompt may need to carry article text, candidate values, normalization rules, and output-format constraints at the same time. We organize biochemical extraction as a sequence of smaller decisions with explicit intermediate representations: the full-text reader produces property evidence, property-specific graph retrieval supplies compact candidate sets, and the final normalizer records an LLM decision for each value. This organization keeps candidate payloads focused and makes the final predictions auditable through cached artifacts.

The research question is where to place LLM reading capacity, graph candidate retrieval, and task decomposition in a practical challenge pipeline. The system described here uses LLMs at the two semantic stages: reading the full article and making final normalization decisions. The knowledge graph provides a local, auditable source of focused property candidates for the normalizer.

3. Task and Dataset

The BiKE/NatUKE task evaluates whether a system can recover hidden property links for a scientific article. We focus on the third-stage evaluation setting used in the submitted result package. The five target properties are:

- **Compound Name**: natural product compounds discussed, isolated, or identified in the article;
- **Bioactivity**: controlled biological activity labels associated with the extracted compounds;
- **Species**: source organisms from which the natural products were obtained;

- **Collection Site**: geographic collection locations for the source material;
- **Type of Isolation**: controlled isolation-type labels.

In the benchmark files, each target relation is represented as a DOI-to-value link. The per-property test CSV files use three columns: `node`, `neighbor`, and `type`. The `node` column is the article DOI, `neighbor` is the target value, and `type` names the relation, such as `doi_name` or `doi_bioActivity`. A system therefore predicts ranked values for each DOI and property.

Throughout this paper, a property means one of the five target fields listed above. A value is a concrete string or label for one property, such as a compound name, `Anticancer`, or `Manaus/AM`. A KG entity is a node label in the benchmark graph. A candidate entity is a KG entity retrieved for a particular extracted value, and a normalized output is the final prediction after preserving, completing, or replacing the extracted value.

The KG files used by the local package contain DOI nodes, nodes for the five target property groups, and auxiliary nodes such as `topic`, `molType`, molecular mass, cLogP, and TPSA. Edges carry an `edge_type` attribute that records the relation between a DOI and a property or auxiliary node, for example `doi_name` and `doi_collectionSite`.

The official property-specific Hits@K settings are $k = 50$ for compound name, $k = 5$ for bioactivity, $k = 50$ for species, $k = 20$ for collection site, and $k = 1$ for isolation type. The local reproduction package is configured for 134 test papers and expects the official benchmark CSV and graph files under `test-data`. Raw article PDFs are expected under `test-raw-pdf`. These files remain external to the public package because the benchmark files and scientific articles have separate licensing and copyright constraints.

4. Method

Figure 1 summarizes the submitted pipeline. It contains four main stages: OCR, full-text extraction, knowledge graph candidate retrieval, and LLM-based entity normalization.

4.1. OCR and Full-Text Extraction

For each target article, the DOI in the BiKE input record identifies the corresponding local PDF file; that PDF is then passed into the OCR process. Each input PDF is converted to Markdown with the Mistral OCR model `mistral-ocr-latest` [4]. The OCR output is treated as the article text for the downstream stages. The `FullTextExtractor` then reads the full OCR text and extracts all five target attributes in one structured JSON object. It also preserves compound-to-bioactivity links when the article explicitly provides them. Here, a compound-to-bioactivity link records which extracted compound is associated with which bioactivity statement in the article.

The extractor also receives a small context block derived from the article DOI. This block is built from local graph neighbors associated with the DOI, including likely topics, molecular types, and physicochemical values. It is used as background information for article reading and disambiguation. The extracted values are still required to be supported by evidence in the article text.

The extractor is configured with `claude-sonnet-4-20250514` [5], temperature 0.0, and an 8192-token output budget. The prompt asks for standard English-compatible spellings, controlled labels for bioactivity and isolation type, and normalized collection-site forms such as `City/ST` when the evidence supports them.

4.2. Property-Isolated KG Candidate Retrieval

The second stage is a local Python retrieval layer that deterministically prepares candidate packages for the normalizer. This retrieval layer uses NetworkX, a publicly available Python library [6], as its main technical support and loads one graph for each property. It builds a candidate index from the allowed graph resources. The graph configuration is property-specific: compound names, bioactivities, species,

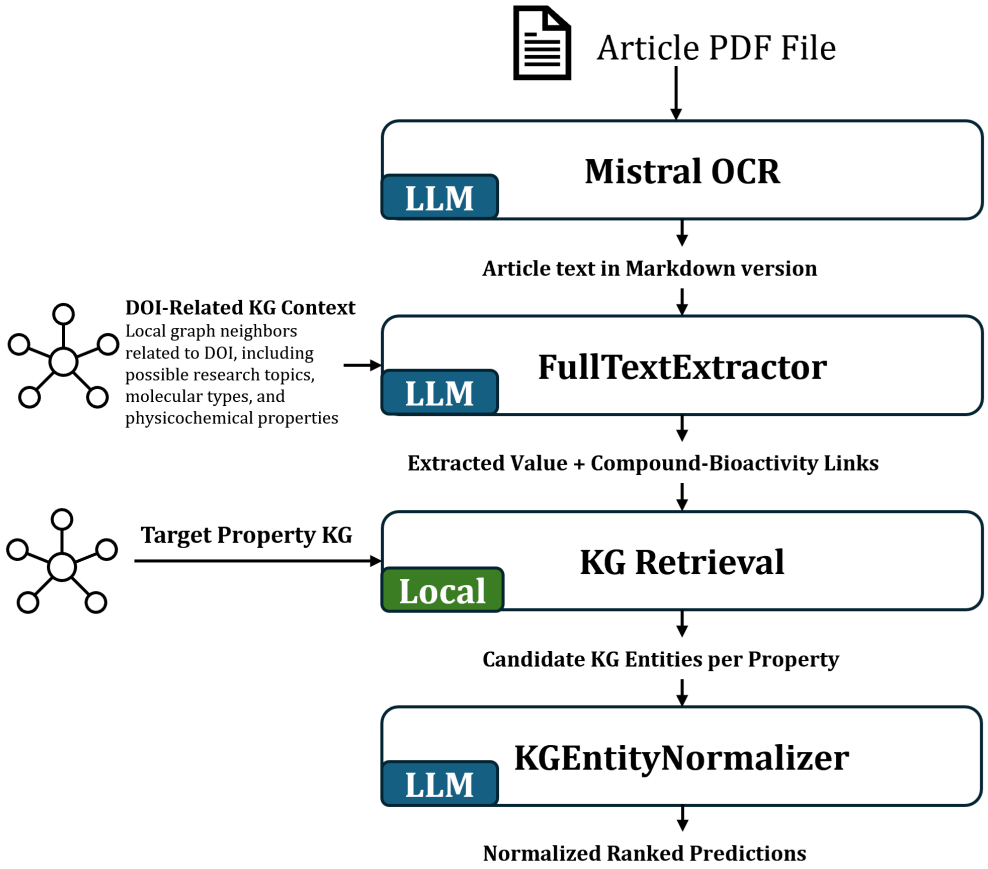

Figure 1: Overview of the submitted BiKE extraction pipeline. An article PDF is converted into Markdown text by Mistral OCR. The `FullTextExtractor` performs LLM-based full-text extraction with DOI-related KG context from local graph neighbors, producing extracted values and compound–bioactivity links. The local KG Retrieval stage uses the target-property KG to generate candidate KG entities per property, and `KGEntityNormalizer` produces normalized ranked predictions.

collection sites, and isolation types are indexed separately. The retrieval layer preserves the property field assigned by the extractor and searches each value only in the corresponding property graph.

For each value emitted by the `FullTextExtractor`, the retrieval layer applies property-specific normalization and searches the corresponding property graph. It records exact string matches, normalized matches, controlled label matches, species family completions, city/state completions, contained entity matches, and fuzzy candidates. Fuzzy matches become weak hints for downstream judgment. When an extracted value cannot be linked to a retrieved KG entity candidate, the payload records this unmatched state explicitly. In this case, the normalizer receives the original extracted value and decides whether to preserve it or leave it unresolved.

This stage is the main mechanism for reducing prompt-side candidate load. The normalizer receives a compact set of candidates tied to a specific extracted value and property.

4.3. LLM Entity Normalization

The `KGEntityNormalizer` receives two inputs: the article-level extraction output and the same-property candidate retrieval package. It is configured with `gpt-4o`, temperature 0.0, and an 8192-token output budget. For every extracted value, it must emit exactly one decision:

- `keep`: preserve the extracted value;
- `complete_with_kg`: use a KG entity that clearly completes a correct incomplete value;

- `replace_with_kg`: replace by a harmless canonicalization of the same entity;
- `drop`: filter a value judged spurious or unsupported from the final prediction set.

The normalizer uses the KG candidates attached to the same extracted value and same property. The final export stage ranks the retained normalized candidates and writes the per-paper prediction files used by the audit script.

5. Experimental Setup

The submitted configuration uses the following models and deterministic settings:

- OCR: `mistral-ocr-latest`;
- Full-text extraction: `claude-sonnet-4-20250514`;
- KG entity normalization: `gpt-4o`;
- temperature: 0.0 for both LLM stages.

The pipeline is implemented as a notebook-first reproducibility package. The intended execution order is: verify benchmark files, verify PDFs, verify API keys, run OCR, run the extractor and normalization stages, and audit the final outputs. Generated runtime directories such as `cache`, `outputs`, and `results` are local artifacts excluded from the public package. The code also contains an audit script that reports exact hit rate, official property-specific Hits@K, stricter local Hits@K, and comparisons against benchmark baselines.

The official benchmark files and article PDFs remain external to the repository. The paper therefore reports the submitted presentation results as the authoritative result source. Method details are cross-checked against the local configuration, prompts, and pipeline code.

6. Results

Table 1 compares the submitted system with legacy NatUKE/BiKE graph baselines and with the BiKE 2025 ensemble result reported in the prior challenge paper [3, 1]. The legacy rows are the benchmark baselines used by the local audit script, and the BiKE 2025 champion row is the SOTA reference used in the BiKE'26 presentation.

Table 1
Main BiKE/NatUKE Hits@K comparison. Source citations are shown in the model column. Legacy rows are graph and embedding benchmark baselines, the BiKE 2025 row is the prior challenge champion result, and the submitted row uses the property-specific official k values: compound 50, bioactivity 5, collection site 20, species 50, and isolation type 1.

Model	Compound @50	Bioactivity @5	Collection Site @20	Species @50	Isolation Type @1
DeepWalk [7]	0.00	0.11	0.43	0.14	0.18
Node2Vec [8]	0.00	0.08	0.33	0.16	0.05
Metapath2Vec [9]	0.09	0.11	0.45	0.43	0.26
EPHEN [10]	0.04	0.66	0.50	0.18	0.75
BiKE 2025 champion [1]	**0.90**	**0.83**	**0.77**	**0.97**	**0.96**
Submitted pipeline	0.89	0.76	0.60	0.89	**0.96**

The legacy baselines show the scale of the shift from graph-only prediction to LLM-centered biochemical reading. EPHEN is the strongest legacy baseline on bioactivity, collection site, and isolation type, and Metapath2Vec is the strongest legacy baseline on compound and species. The submitted pipeline is above all four legacy baselines on every property. The local audit reports macro-average official Hits@K values of 0.172 for DeepWalk, 0.124 for Node2Vec, 0.268 for Metapath2Vec, 0.425 for EPHEN, and 0.819 for the submitted pipeline.

The strongest submitted result is isolation type, where the system matches the BiKE 2025 champion result at 0.96 Hits@1. Compound extraction is also close to the BiKE 2025 champion result: 0.89 versus 0.90 at Hits@50. This is important because compound names are numerous, chemically variable, and sensitive to OCR and spelling differences. The result suggests that the initial full-text reader plus KG-assisted canonicalization preserves many correct compound candidates with a compact first prompt.

Bioactivity reaches 0.76 Hits@5, compared with 0.83 for the BiKE 2025 champion and 0.66 for EPHEN. Bioactivity extraction is constrained by a controlled label set, and its main difficulty is associating each activity with the compounds reported as article findings. Species reaches 0.89 Hits@50, a strong absolute score compared with the best legacy baseline of 0.43 and the BiKE 2025 champion value of 0.97. The remaining gap may reflect taxonomic completion errors, OCR damage, and cases with low-confidence family-level graph detail.

Collection site is the main weakness, with 0.60 Hits@20 versus the SOTA reference of 0.77, while remaining above the strongest legacy baseline value of 0.50. This property combines textual ambiguity with strict normalization. Articles may mention countries, reserves, campuses, city names, Brazilian state abbreviations, and narrative locations. The pipeline normalizes to `City/ST` or `N/A/ST`. Wrong city-state completion and overly conservative normalization can reduce Hits@K.

The presentation also reports stricter candidate settings for the submitted pipeline: compound Hits@15 is 0.87, bioactivity Hits@5 is 0.73, collection site Hits@5 is 0.58, species Hits@5 is 0.88, and isolation type Hits@1 is 0.91. The small drop from compound Hits@50 to Hits@15 suggests that many correct compound predictions are ranked relatively early. Isolation type remains sensitive because it is evaluated at $k = 1$.

7. Error Analysis

We further inspected the local audit artifacts for the final exported run. The audit contains 282 property-level comparisons across the 134-paper subset, with 231 hits and 51 misses. The rounded per-property rates match the submitted presentation values: 0.89 for compound name, 0.76 for bioactivity, 0.60 for collection site, 0.89 for species, and 0.96 for isolation type. The remaining errors are highly concentrated. Collection site accounts for 22 misses and bioactivity accounts for 14, so these two properties explain 36 of the 51 misses.

Table 2
Error categories in the final local audit. Categories were assigned from the gold values, exported predictions, and cached stage outputs.

Error class	Count	Main property	Interpretation
Conservative location output	12	Site	The system emits `N/A` when the gold answer expects a concrete city and state.
Wrong geographic normalization	8	Site	The system selects the wrong city/state pair or the wrong `N/A/ST` state.
Bioactivity label confusion	12	Bioactivity	The output is usually a valid controlled label, but it corresponds to the wrong assay or biological claim.
Surface/canonicalization mismatch	7	Mixed	Predictions are near the gold string but fail exact matching because of spelling, authority, or chemical-form differences.
Empty final prediction	3	Bioactivity/Site	The reader or downstream export emits no value for a property with a gold answer.
Species/source selection	3	Species	The system selects a related organism, incomplete taxonomic form, or partial multi-species answer.
Isolation-type confusion	1	Isolation Type	The predicted article source class conflicts with the benchmark's semisynthesis label.
KG candidate available but unused	1	Species	A matching KG candidate appears in the retrieval payload but is not selected in the final output.

Collection site is the clearest remaining bottleneck. More than half of the site misses are conservative decisions in which the system outputs `N/A`, while the benchmark expects a specific Brazilian city/state value such as Araraquara/SP, Belem/PA, Sao Carlos/SP, or Sao Paulo/SP. Another group of errors comes from wrong geographic normalization: for example, a paper-level location mention may be mapped to the wrong city in the same state, or a city-level prediction may be returned when the benchmark

expects an N/A/ST state-level answer. These cases indicate that collection-site performance depends on document-level evidence discrimination. The model must distinguish source-material collection sites from institutional addresses, assay locations, author affiliations, and narrative geographic context.

Bioactivity errors have a different shape. Most misses are controlled-label selection errors. The exported values are often valid KG labels, such as Cytotoxic, Anticancer, Antioxidant, or Inhibition of Glycosidase. The selected label belongs to another assay or biological claim in the paper. Examples include Anticancer being predicted for a Cytotoxic gold label, Cytotoxic being predicted for an Anticancer gold label, and Antichagasic being predicted for an Antitrypanosomal gold label. This suggests that bioactivity performance is primarily limited by evidence association: the system must connect the right assay statement to the right compound findings and ignore background biological context.

The remaining compound and species errors are fewer but instructive. Compound misses include near-string differences such as caffeoate versus caffeate and Fumiquinazoline versus fumiquinozoline. Additional compound misses occur when the predicted form is a related parent compound or derivative and the gold form is a more specific chemical expression. Species misses often involve taxonomic completion or source selection: authority names and family suffixes may differ from the gold form, and multi-species gold answers may be only partially recovered. Isolation type is comparatively stable; the two misses in the local audit involve a semisynthesis-vs-plant-isolated ambiguity and a multi-label gold answer where only one label was returned.

This analysis clarifies the most useful next improvements. Collection-site handling needs stronger location evidence selection and calibrated city/state/N/A/ST normalization. Bioactivity needs explicit compound–activity evidence linking before the controlled label is chosen. Compound and species errors would benefit from a post-normalization canonicalization pass that handles near-string chemical spelling, taxonomic authority variants, and partial multi-value outputs.

8. Ablation and Discussion

Table 3 summarizes the additional routes explored during system development. These runs provide diagnostic analysis of cost, configuration, and normalization stability around the final pipeline.

Table 3
Development routes discussed as ablations. Columns C, B, L, S, and T denote compound, bioactivity, collection site, species, and isolation type. The submitted official-k row uses $C@50$, $B@5$, $L@20$, $S@50$, and $T@1$. The submitted stricter-k row uses $C@15$, $B@5$, $L@5$, $S@5$, and $T@1$. Development-route rows use their own diagnostic k profiles, so the values provide evidence about each route.

Route	C	B	L	S	T	Interpretation
Submitted, official k	0.89	0.76	0.60	0.89	0.96	Main submitted pipeline: strong compound and isolation-type performance with focused KG candidate prompts.
Submitted, stricter k	0.87	0.73	0.58	0.88	0.91	Reduced candidate budget; compound and species remain robust.
Weak-model decomposition	0.754	0.707	0.545	0.889	0.981	Finer task splitting with a weaker model gave reasonable results with higher runtime and lower average performance.
Evidence-review route	0.426	0.879	0.800	0.389	0.926	Evidence filtering plus strong-model review produced high scores on two properties with weaker normalization and output stability.

The comparison with the BiKE 2025 champion system [1] is especially informative from a prompt-design perspective. The reported method used prompt-side entity enumeration: its zero-shot proprietary-LLM ensemble placed article text together with explicit possible target values, including compound-name candidates and controlled property/entity lists, in the prompt. The submitted pipeline uses a different prompt allocation. The full-text extractor receives article text and compact extraction rules, while the KG entity space is handled by a local retrieval layer. The normalizer then receives bounded same-property candidates attached to the extracted values. This organization keeps each LLM call centered on article evidence and a compact retrieval payload; prompt length is controlled by extracted evidence and top-k retrieval depth while the full KG can grow outside the LLM context. In large-scale applied KG settings, enumerating all possible entities in the prompt can make prompt length expand rapidly, increasing

cost, latency, and context-management pressure. The submitted results show that placing a smaller amount of KG-internal information in prompts can still preserve strong compound and isolation-type performance.

The weak-model route supports a nuanced conclusion. Decomposition can make the task easier for smaller models by narrowing each prompt, and model allocation still matters. Some stages require broad article understanding, chemical terminology, and careful normalization. Moving too much responsibility to a lower-capacity model can reduce cost per call and increase the number of calls, total latency, and error propagation.

The evidence-review route explores stronger evidence selection with a scientific-text encoder; SciBERT is a standard encoder for scientific text [11]. In this development route, an evidence filter and strong-model review produced high scores for bioactivity and collection site. Final output stability remained weaker than the submitted pipeline. In particular, content judgment and result-format control showed different failure modes. This distinction is central for BiKE: a system needs accurate article reading and values returned in the form expected by the knowledge graph evaluation.

The main design lesson is that task splits should follow the natural error boundaries of the problem. OCR and article reading form one boundary, graph candidate retrieval forms a second boundary, semantic normalization forms a third boundary, and final export forms a fourth boundary. This design makes failures easier to audit and reserves strong models for stages where semantic judgment is most important.

9. Limitations

The submitted pipeline has several limitations. First, collection site remains difficult. The current rules handle common city/state completions, but they also location phrases are often ambiguous and may require stronger geocoding, document-level evidence selection, or challenge-specific location normalization.

Second, OCR quality directly affects all downstream stages. Chemical names, species names, and accented location strings can be damaged by PDF conversion. The graph retrieval layer contains normalization rules for some common corruptions, and severe OCR damage can still create candidates that are too far from the correct entity.

Third, fuzzy graph retrieval adds useful recall together with extra judgment requirements. It can rescue spelling variants and incomplete names and can also surface plausible wrong candidates. The submitted normalizer treats fuzzy candidates as weak signals, and a more systematic calibration of match thresholds would likely improve reliability.

Fourth, multi-stage execution increases latency and API cost. The approach reduces the amount of KG candidate information placed in any one prompt and adds orchestration overhead with at least two model calls per article after OCR.

Finally, full reproduction requires the user to provide official benchmark files and article PDFs locally. Licensing constraints make the public repository a reproducibility package for scripts, prompts, expected layout, and audit tooling.

10. Conclusion and Future Work

This paper presented a task-decomposed LLM and knowledge graph alignment pipeline for BiKE biochemical property extraction. The system separates full-text article reading, property-isolated KG candidate retrieval, and LLM-based normalization. On the submitted third-stage comparison, it achieves near-SOTA compound extraction, matches SOTA for isolation type, and remains competitive on bioactivity and species with more focused prompt-side KG information. The largest remaining gap is collection site normalization.

Future work should strengthen evidence selection before extraction, improve location normalization, and allocate models more systematically across stages. The error analysis suggests three concrete

priorities: a location resolver that separates collection evidence from affiliation and assay context, a compound–activity linker that verifies which assay labels belong to which reported compounds, and a final canonicalization pass for near-correct chemical and taxonomic strings. The development routes suggest that strong models are most valuable for final article understanding and normalization. Lighter components may be useful for evidence filtering and candidate pruning. A cost-aware verifier that combines graph constraints, source-text evidence, and calibrated LLM decisions is a promising next step.

Acknowledgements

We thank the BiKE organizers for maintaining the benchmark and evaluation setting, and the NatUKE authors for making the underlying task and code available to the community.

Declaration on Generative AI

During the preparation of this work, the author(s) used GPT-5.4 in order to: Grammar and spelling check. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] M. P. S. Gôlo, J. G. B. de Medeiros Junior, G. S. V. Boas, F. M. F. Lobato, D. F. Silva, R. M. Marcacini, Large language models ensemble for biochemical properties detection in scientific articles, in: Proceedings of the Second International Biochemical Knowledge Extraction Challenge (BiKE 2025), co-located with Text2KG at ESWC 2025, volume 4020 of *CEUR Workshop Proceedings*, 2025, pp. 215–226. URL: https://ceur-ws.org/Vol-4020/BiKE_Paper_ID_1.pdf.

[2] E. Marx, M. Valli, J. da Silva e Silva, P. V. do Carmo, Preface of the second international biochemical knowledge extraction challenge (bike), in: Proceedings of the Second International Biochemical Knowledge Extraction Challenge (BiKE 2025), co-located with Text2KG at ESWC 2025, volume 4020 of *CEUR Workshop Proceedings*, 2025. URL: https://ceur-ws.org/Vol-4020/BiKEPreface_2025.pdf.

[3] P. V. do Carmo, E. Marx, R. M. Marcacini, M. Valli, J. V. S. e Silva, A. Pilon, Natuke: A benchmark for natural product knowledge extraction from academic literature, in: 2023 IEEE 17th International Conference on Semantic Computing (ICSC), IEEE, 2023, pp. 199–203. doi:10.1109/ICSC56153.2023.00039.

[4] Mistral AI, Mistral OCR, 2025. URL: https://mistral.ai/news/mistral-ocr/, accessed: 2026-06-19.

[5] Anthropic, System card: Claude Opus 4 and Claude Sonnet 4, 2025. URL: https://www-cdn.anthropic.com/4263b940cabb546aa0e3283f35b686f4f3b2ff47/claude-opus-4-and-claude-sonnet-4-system-card.pdf.

[6] A. A. Hagberg, D. A. Schult, P. J. Swart, Exploring network structure, dynamics, and function using NetworkX, in: Proceedings of the 7th Python in Science Conference, 2008, pp. 11–15.

[7] B. Perozzi, R. Al-Rfou, S. Skiena, DeepWalk: Online learning of social representations, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2014, pp. 701–710. doi:10.1145/2623330.2623732.

[8] A. Grover, J. Leskovec, node2vec: Scalable feature learning for networks, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 855–864. doi:10.1145/2939672.2939754.

[9] Y. Dong, N. V. Chawla, A. Swami, metapath2vec: Scalable representation learning for heterogeneous networks, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2017, pp. 135–144. doi:10.1145/3097983.3098036.

[10] P. do Carmo, R. Marcacini, Embedding propagation over heterogeneous event networks for link prediction, in: 2021 IEEE International Conference on Big Data (Big Data), 2021, pp. 4812–4821. doi:10.1109/BigData52589.2021.9671645.

[11] I. Beltagy, K. Lo, A. Cohan, SciBERT: A pretrained language model for scientific text, in: K. Inui, J. Jiang, V. Ng, X. Wan (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3615–3620. URL: https://aclanthology.org/D19-1371/. doi:10.18653/v1/D19-1371.

[12] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-augmented generation for knowledge-intensive nlp tasks, in: Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS '20, Curran Associates Inc., Red Hook, NY, USA, 2020.

[13] D. Zhou, N. Scharli, L. Hou, J. Wei, N. Scales, X. Wang, D. Schuurmans, O. Bousquet, Q. Le, E. H. Chi, Least-to-most prompting enables complex reasoning in large language models, ArXiv abs/2205.10625 (2022). URL: https://api.semanticscholar.org/CorpusID:248986239.

[14] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. H. Chi, Q. V. Le, D. Zhou, Chain-of-thought prompting elicits reasoning in large language models, in: Proceedings of the 36th International Conference on Neural Information Processing Systems, NIPS '22, Curran Associates Inc., Red Hook, NY, USA, 2022.